Вестник УГАТУ
Уфа : УГАТУ, 2012 Т. 16, № 3 (48). С. 138–142

УДК 519.7

А. А. Свенч, Р. Т. Файзуллин, И. Г. Хныкин

ГИБРИДНАЯ СУПЕРКОМПЬЮТЕРНАЯ СИСТЕМА

В статье рассматривается гибридная суперкомпьютерная система на базе кластера центральных процессоров (ЦПУ), кластера графических процессоров (ГПУ) и системы хранения данных (СХД). Описывается программно-аппаратная база, используемая для построения. Формируется класс задач, которые могут оптимально выполняться с использованием гибридной системы. Рассматриваются прикладные задачи алгебры, криптографического анализа, моделирования транспортных потоков в сложных системах. *Суперкомпьютер*; *центральный процессор*; *графический процессор*; *алгебра*; *криптография*; *транспорт*; *автомобилестроение*; *химическая физика*

В настоящее время есть множество общедоступных вычислительных кластеров (суперкомпьютеров). Одни основаны на стандартной архитектуре CPU, другие на менее известной архитектуре GPU. Проведенный обзор показал, что вычислительных систем, объединяющих данные архитектуры в единый кластер, нет, либо они являются коммерческим секретом.

Архитектура CPU позволяет использовать практически любые шаблоны параллельных вычислений, в то время как GPU является менее гибким, но гораздо более производительным вариантом для решения многих вычислительных задач. В частности, архитектура современных GPU Nvidea Tesla включает в себя множество масштабируемых блоков, не обладающих мощной управляющей логикой и большим объемом кэш-памяти. Эта архитектура может эффективно применяться при вычислениях с большим параллелизмом и интенсивной арифметикой. Все функции, выполнимые на GPU, не поддерживают рекурсии и имеют некоторые другие ограничения, которых нет в архитектуре CPU. С помощью созданной авторами модели программирования стало возможным разрабатывать сложные проекты с использованием параллельных вычислений, которые одновременно могут использовать и гибкость CPU и более скоростные вычисления GPU.

Другим недостатком существующих суперкомпьютерных систем является непрозрачный доступ пользователя к своим проектам на кластере. Сегодня пользователю предлагается обучиться работать с арсеналом программного обеспечения (программой удаленного доступа, командной строкой, компилятором, планировщиком и т.д.), установленным на целевом кластере. Поэтому авторами разработана программная система, позволяющая осуществлять удаленную работу с проектами без необходимости погружаться в устройство операционной системы кластера.

ОПИСАНИЕ ТЕХНОЛОГИИ И АППАРАТНЫХ СРЕДСТВ

Гибридная суперкомпьютерная система включает в себя:

- Вычислительную систему, объединяющую кластер CPU(центральных процессоров) x86/64 – на базе процессоров Intel Xeon, и кластер GPU(графических процессоров) – на базе процессоров NVidia Tesla.
- Модуль системы хранения данных на базе оборудования Sun Microsystems.
- Инновационную модель программирования, позволяющую объединить вычисления, использующие CPU и GPU.
- Инновационную программную систему, которая управляет выполнением проекта, использующего гибридные вычисления, и предоставляет прозрачный интерфейс для работы пользователей с системой на базе web-технологий.

На сегодняшний день в распоряжении ОмГТУ находится пять вычислительных узлов с архитектурой CPU (mgr, cn01, cn02, cn03, cn04). Каждый вычислительный узел включает два 4-ядерных процессора HP X5472 DL 160G5.

Контактная информация: 8(3812) 65-20-93

Работа поддержана грантом конкурса "У.М.Н.И.К", конференция "Омское время – взгляд в будущее", 2010 г.

Статья рекомендована к публикации программным комитетом международной научной конференции "Параллельные вычислительные технологии 2011"

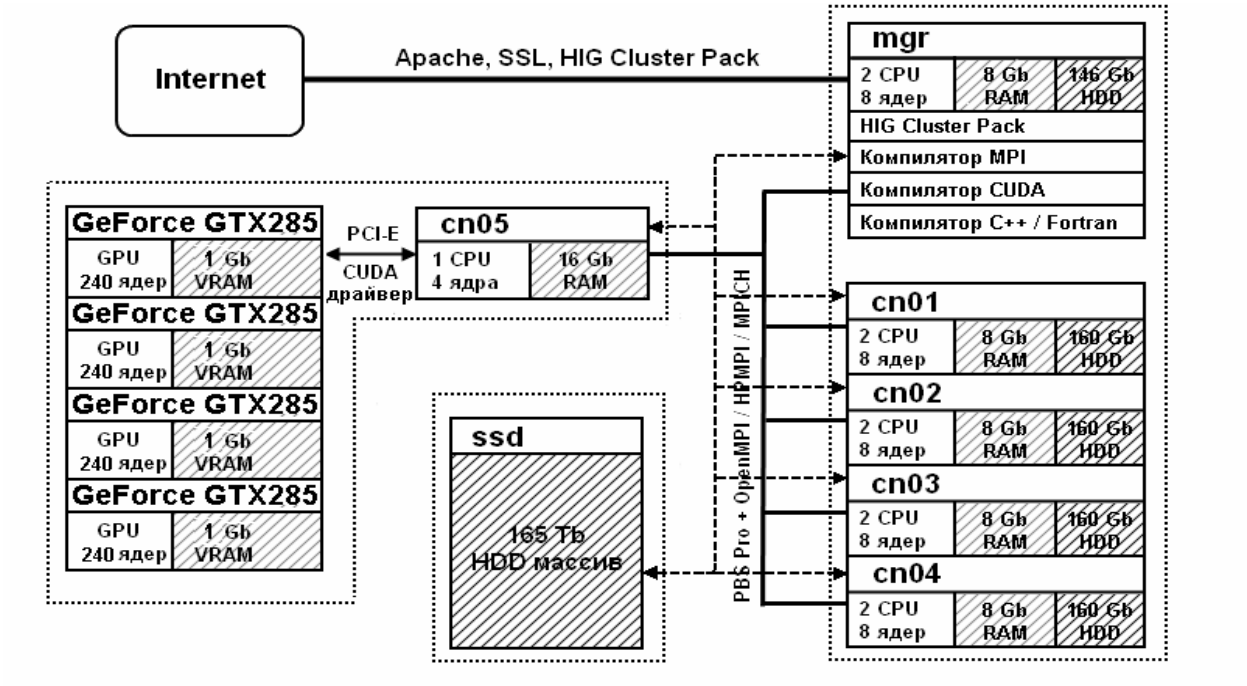

Рис. 1. Схема суперкомпьютерной системы ОмГТУ

Пиковая производительность достигает 1 Tflop, а объем оперативной памяти составляет 40GB. Структура масштабируема и расширяема. Вычислительная мощность может быть увеличена за счет добавления дополнительных узлов.

В рамках гибридизации к суперкомпьютеру добавлен вычислительный узел cn05, осуществляющий управление кластером GPU. Для его построения выбрана платформа NVidia Tesla 10 с архитектурой CUDA GPU. Узел представляет собой персональный компьютер с подключенными к нему ячейками NVidia Tesla S1070, каждая из которых включает в себя 4 GPU с суммарной пиковой вычислительной мощностью 4 Tflops в операциях с одинарной точностью. То есть производительность всей системы, при относительно малых затратах увеличивается до 5 Tflops (при задействовании одной ячейки Tesla S1070). Для выполнения параллельных вычислений используется NVidia CUDA API для языков программирования C, C++, Fortran [2, 3].

Другим дополнением суперкомпьютерной системы является модуль хранения данных (ssd) на базе системы Sun StorageTek 9900V. Данная система отличается высокой отказоустойчивостью.

Все узлы объединены в локальную высокоскоростную сеть (1Gb/сек). Параллельное выполнение программ осуществляется с помощью технологий MPI и CUDA. Поддерживаемые реализации MPI: OpenMPI, HPMPI, MPICH [1].

Для управления кластером один из его узлов выделен как управляющий (mgr), при необходимости он может быть использован для вычислений. На управляющем узле установлена авторская программная система для управления проектами на кластере (HIG_Cluster_Pack). С ее помощью узел принимает и выполняет команды авторизованных пользователей через Интернет. Все соединения осуществляются по защищенному HTTPS протоколу (см. рис. 1).

ПРОГРАММНАЯ СИСТЕМА ПРОЗРАЧНОГО ДОСТУПА К КЛАСТЕРУ

Программная система доступа к кластеру разработана авторами проекта и основана на базе web-сервера Apache. Данная система является инновационной полностью переносимой и расширяемой и может работать на вычислительных кластерах с любой аппаратной конфигурацией, операционной системой и программным обеспечением. На сегодняшний день данная система (HIG_Cluster_Pack) не имеет аналогов и успешно используется на кластере ОмГТУ.

Программа предоставляет следующие возможности:

- Трехфакторная система авторизации на уровне web-сервера, самой программы и операционной системы.
- Для каждого пользователя создается ограниченная зона выполнения данных. То есть, все исполняемые файлы проекта могут обращаться только к ресурсам своего проекта и таким образом исчезает возможность злонамеренного повреждения кластера или проектов других пользователей.
- Пользователю предоставляется удобный web-интерфейс, который позволяет создать / удалить проект, загрузить проект, скомпилировать проект, запустить проект, посмотреть файлы проекта и т. п. При этом поддерживаются все необходимые настройки работы с проектом. Пользователю нет необходимости разбираться в опциях компилятора и планировщика заданий.
- Существует система полуавтоматической обработки заданий, когда пользователь с помощью определенных HTTPS запросов управляет работой проекта на кластере. Таким образом, появляется возможность создания программ, работающих на персональном компьютере и использующих вычислительные мощности кластера для просчета ресурсоемких блоков.
- Благодаря дружественному интерфейсу регистрация нового пользователя производится удобным как самому пользователю, так и администратору кластера способом. Пользователю достаточно сформировать запрос на сертификат доступа, где указываются необходимые данные. Администратору достаточно разрешить работу на кластере с данным сертификатом. Все остальные действия выполняются автоматически.

ПРИКЛАДНЫЕ ЗАДАЧИ

Суперкомпьютерная система, смоделированная авторами в виде рабочего прототипа, ориентирована на широкий круг задач. В сочетании с инновационными авторскими разработками по созданию унифицированных средств доступа и управления, данная система позволяет использовать сразу все ведущие решения в области высокопроизводительных вычислений.

Математическое моделирование транспортных потоков на основе микроскопической схемы «предиктор-корректор»

Представляется возможным использование параллельных вычислений в задаче моделирования транспортных потоков, которая отличается большой сложностью в случае моделирования и оптимизации работы транспортной сети крупного города. Существующие модели, макроскопические и микроскопические, в настоящее время ограничены однопроцессорными реализациями.

В качестве топографической основы модели транспортной системы города рассмотрим неориентированный граф, ребра – это дороги или магистрали, узлы – это перекрестки. Будем считать, что движение везде двухполосное, разделенное сплошной линией, т. е. обгоны запрещены и на всех перекрестках стоят светофоры. Транспортные средства – это точки, расположенные на ребрах. В начальном состоянии системы все транспортные средства имеют нулевую скорость. Для каждого транспортного средства случайно выбираются пункты назначения – точки на каком-либо ребре, для которых вычисляется оптимальный маршрут. В результате поиска пути, каждому транспортному средству с номером i_s, поставлен в соответствие массив номеров ребер $ir_s(l)$, которые он должен пройти. Здесь $l = 1,..,L$, где L – это число ребер в графе. Время считаем изменяющимся дискретно и вводим два характерных величины для времени dt и du, где первая величина намного больше второй. Мы предполагаем, что dt – это общий интервал времени, на которое независимо прогнозируется движение каждым водителем, а du – это шаг по времени после которого водитель вынужден корректировать свой прогноз.

Решение частных задач моделирования и оптимизации естественным образом приводит нас к требованию максимального повышения производительности вычислений. Как показывают расчеты, движение в режиме реального времени транспортных средств в количестве 100 000 единиц моделируется одним процессором, но задачи моделирования большей размерности и особенно оптимизационные задачи требуют повышения скорости вычислений на порядки.

Например, если речь идет об управлении движением транспорта, то число расчетов по необходимости должно быть велико, и они должны происходить существенно быстрее.

Как показывает опыт вычислений, число контролируемых транспортных средств при подобном моделировании может достигать чисел порядка 106 и время расчета вариантов движения существенно меньше, чем время реализаций этих вариантов в действительности. Также,

представляется перспективным перенести часть массовых вычислений на графические процессоры, выбирая оптимальные размеры региона или специальным образом распараллеливать вычисления на обычные и графические процессоры [4].

Решение больших систем линейных алгебраических уравнений с ленточной положительно определенной матрицей

Еще одна из прикладных задач, которую можно оптимизировать, используя гибридную систему – решение больших (размерности 10^8 и более) систем линейных алгебраических уравнений с ленточной положительно определенной матрицей итерационным методом. Эта задача возникает при решении многих задач математической физики. Несмотря на алгоритмическую простоту решения, реализация решения этой задачи на ЭВМ при больших размерностях системы вызывает определенные затруднения, связанные с размещением в памяти и последующим доступом к данным системы в памяти устройства.

Для эффективного решения этой задачи используются следующие приемы:

1. Хранение исходной матрицы коэффициентов СЛАУ в «разреженном» формате. По условиям, матрица коэффициентов имеет ленточную структуру, т. е. лишь малая часть коэффициентов будут ненулевыми. Это делает целесообразным хранение только ненулевых коэффициентов при загрузке системы в память.

2. Решение большой системы по принципу «разделяй и властвуй» путем решения отдельных ее блоков с «перехлестом», получаемых на каждой итерации векторов решения. Все блоки при этом будут одинаковой размерности N, которая выбирается исходя из объемов памяти устройств. В силу диагональной структуры матрицы коэффициентов СЛАУ разделение на блоки является тривиальной задачей. После каждой итерации решения «блока» вектор его решения корректируется векторами решений соседних «блоков» (по участкам, где используется «перехлест»).

3. Для решения блока используется метод Якоби, который дает приемлемую сходимость (с учетом диагонального преобладания матрицы коэффициентов исходной СЛАУ), и не требует обновления данных «на лету», как метод Гаусса-Зейделя.

Итерационный процесс метода Якоби можно распараллелить на большое количество вычислительных ядер, каждое из которых работает независимо. Это делает данный этап вычислений хорошим кандидатом для выполнения на GPU. Итерация выполнения разбивается на перемножение матрицы коэффициентов на текущий вектор решения, формирование нового вектора решения, вычисление векторов невязки решения и системы. Каждое из этих действий может быть выполнено посредством программы на GPU.

4. Для конкретной реализации алгоритма решения системы необходимо определить оптимальный размер блоков (N), на которые производится разбиение системы уравнений, величину области перехлеста блоков (H).

Величина N выбирается исходя из требований к памяти. Для выбранной аппаратной конфигурации оптимальным оказывается значение $N = 4096$. При этом подразумевается, что количество ненулевых элементов в строке матрицы не превосходит 2048. Значение H для такой размерности блоков – 256 (в случае, если удвоенное количество ненулевых элементов в строке исходной матрицы коэффициентов не превосходит этого значения, иначе – ближайшая сверху к указанному числу степень двойки).

Нами было разработано и протестировано две реализации описанного алгоритма, каждая из которых использует свою модель распараллеливания. В реализации для CPU все шаги алгоритма, кроме самого первого, осуществляются в цикле MPI-команд. Реализация версии с использованием гибридных вычислений использует CUDA для решения блоков, остальные вычисления и передачи данных производятся на ядрах CPU с использованием MPI.

Гибридная версия алгоритма показала уменьшение времени решения по сравнению с реализацией алгоритма с помощью MPI в среднем более чем в 20 раз. Например, время решения пяти диагональной СЛАУ размерностью 14×10^6 гибридной версией алгоритма составляет 16 часов.

Применение гибридного суперкомпьютера для решения других прикладных задач

Предложенная в статье технология применяется для увеличения скорости решения большого числа прикладных задач. Например, в [4] описаны алгоритмы и результаты применения

гибридной вычислительной системы к решению задач криптографического анализа ассиметричных шифров и задач алгебры (разложение чисел на простые сомножители, дискретное логарифмирование, дискретное логарифмирование на эллиптической кривой). В [6] гибридный суперкомпьютер используется для решения задач оптимального управления перемещением объекта в трехмерном пространстве с препятствиями. В [7] предлагается методика моделирования адсорбции сложных нелинейных молекул на квадратную решетку, которая также использует гибридный суперкомпьютер для проведения ресурсоемких вычислений.

СПИСОК ЛИТЕРАТУРЫ

1. Материалы сайта http://www.mpi-forum.org Стандарт MPI-2.0.

2. Материалы сайта http://developer.download.nvidia.com Комплект документации по CUDA.

3. Материалы сайта http:// developer.nvidia.com Комплект тестовых проектов CUDA

4. **Файзуллин Р. Т., Свенч А. А., Хныкин И. Г.** Применение гибридной суперкомпьютерной системы в задачах криптоанализа // Доклады ТУСУР. 2010. № 1 (21), ч. 1. С. 61–63.

5. Potential-induced phase transition of trimesic acid adlayer on Au(111) / G. J. Su [et al.] // J. Phys. Chem. B. 2004. V. 108. P. 1931–1937.

6. **Корытов М. С.** Построение матрицы смежности графа поверхности с препятствиями для поиска кратчайшей траектории перемещения груза автомобильным краном // Матер. 69-й Международн. науч.-техн. конф. Ассоциации автомобильных инженеров (ААИ). Омск: СибАДИ, 2010

7. **Мышлявцев А. В., Стишенко Л. Г.** Фазовые переходы в модели адсорбции сложных нелинейных молекул на квадратную решетку: метод Монте-Карло // Современная химическая физика: 20 симпозиум, 15–26 сентября, Туапсе. М., 2008. С. 147–148.

ОБ АВТОРАХ

Свенч Андрей Александрович, асп. Омск. гос. техн. ун-та. Иссл. в обл. параллельных вычислений.

Файзуллин Рашит Тагирович, проф. Омск. гос. техн. ун-та. Д-р техн. наук. Иссл. в обл. вычислительн. математики, криптографии, криптоанализа.

Хныкин Иван Геннадьевич, доц. Омск. гос. техн. ун-та. Канд. физ.-мат. наук. Иссл. в обл. вычислительн. математики, инф. безопасности, криптографии, криптоанализа.